Big Data Applications in TURKSTAT

Digital Transformation and Projects Department
Big Data Group
Outline

- TurkStat Big Data Group
- Project 1: Enhancing Performance of Rule-based COICOP Assignation by Using Big Data Tools
- Project 2: COICOP Assignation to Web Scrapped Products with ML-based Approach
  - Background
  - Data collection & deduplication phase
  - Assigning COICOP with ML Techniques
  - Assigning COICOP with DL Techniques
- Project 3: Assigning ISCO codes to Job Advertisements
  - Background
  - Data collection
  - Assigning ISCO codes with ML Techniques
  - Assigning ISCO codes with DL Techniques
TurkStat Big Data Group

Several project in development

- Working on big data infrastructure
- Using machine learning/deep learning algorithms
- Based on different data sources (web scraping, scanner data)
- Both batch processing and stream processing
Projects consists of these stages

- Collection of daily product and online job data from up to 90 web sites
- Preprocessing, deduplication and classification of daily products by using Natural Language Processing (NLP)
- Developing machine learning and deep learning models to perform time series analysis to identify the effects of prices changes between product groups in terms of time interval and ratio.
- Developing machine learning and deep learning models to classify jobs and skills based on online job vacancy data.
- Applying developed models on daily collected data to perform lag analysis on product groups in CPI basket.
- Applying developed models on daily collected online job vacancy data to visualize jobs and skills demands based on sectors, locations and times.
Project 1: Enhancing Performance of Rule-based COICOP Assignation by Using Big Data Tools

- Previously implemented via Python programming language in a traditional way
- 150K code assignation daily with this way
- Re-implemented with Apache Spark runs on 16-nodes Hadoop cluster
**Project 1: Enhancing Performance of Rule-based COICOP Assignation by Using Big Data Tools**

- 4M code assignment daily with this approach

- Thanks to this approach code assignment process which takes 200 days previously can be completed in 7 days (for 30M rows dataset).

- This time can be further reduced by adding new nodes to the existing Hadoop cluster.
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Background

• Rule-based code assignation processes are not maintainable

• When new products are received, new rules may be required.
**Project 2: COICOP Assignment to Web Scraped Products with ML-based Approach**

**Dataset**

- There are 3 main sources.
  - Web scraping data
  - Scanner data
  - Survey data (from Turkstat regional offices)

- Scanner data and survey data are labelled

- Main Columns: Product definition, COICOP code (if assigned)
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Dataset

The problem is how to save web scraped data to the cluster.
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Related Spark Code

```python
ssc = StreamingContext(sc, 2)
price_kvs = KafkaUtils.createDirectStream(ssc,
        config['price_topics'],
        config['brokerList'],
        valueDecoder=lambda x: x)

price_lines = price_kvs.map(lambda x: (x[1]))
price_lines.foreachRDD(donecessary4Price)
ssc.start()
ssc.awaitTermination()
```
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach
### Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

- Spark codes reads data from Kafka topics, do some preprocessing jobs on streaming data and save to Hive tables.

<table>
<thead>
<tr>
<th>website</th>
<th>product_name</th>
<th>product_price</th>
<th>product_code</th>
<th>category</th>
<th>normalized_extra_info</th>
<th>singular_product_name</th>
<th>unique_product_name</th>
<th>amount</th>
<th>unit</th>
<th>brand</th>
<th>coicop_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>cagri/hipermarket</td>
<td>Koroplast Doğada Çözülebilen Çop Torbasi 10lu</td>
<td>9.95</td>
<td>10098</td>
<td>market</td>
<td>koroplast.adet</td>
<td>doğada çözülebilen çop torbasi</td>
<td>koroplast doğada çözülebilen çop torbasi 10</td>
<td>10</td>
<td>adet</td>
<td>koroplast</td>
<td>5612030401</td>
</tr>
<tr>
<td>cagri/hipermarket</td>
<td>Knorr Kremalı Mantarı Makarna Sosu</td>
<td>4.95</td>
<td>1773</td>
<td>market</td>
<td>knorr.adet</td>
<td>kremalı mantarı makarna sosu</td>
<td>knorr kremalı mantarı makarna sosu</td>
<td>1</td>
<td>adet</td>
<td>knorr</td>
<td>111501</td>
</tr>
<tr>
<td>cagri/hipermarket</td>
<td>Torku Süt Kaçaolu 6x180 ml</td>
<td>8.95</td>
<td>7644</td>
<td>market</td>
<td>torku.sut.adet</td>
<td>süt kaçaolu</td>
<td>torku süt kaçaolu 1080.0 millilitre</td>
<td>1000</td>
<td>millilitre</td>
<td>torku süt</td>
<td>114101</td>
</tr>
<tr>
<td>cagri/hipermarket</td>
<td>İçimino Çikolata Süt 6x200 ml</td>
<td>7.95</td>
<td>3924</td>
<td>market</td>
<td>icim.adet</td>
<td>cikolata süt</td>
<td>içimino cikolata süt 1200.0 millilitre</td>
<td>1200</td>
<td>millilitre</td>
<td>içim</td>
<td>114101</td>
</tr>
<tr>
<td>cagri/hipermarket</td>
<td>Tamek Reçel Böğürlen 380 gr</td>
<td>11.45</td>
<td>10308</td>
<td>market</td>
<td>tamek.adet</td>
<td>reçel bogurten</td>
<td>tamek reçel bogurten 380.0 gram</td>
<td>380</td>
<td>gram</td>
<td>tamek</td>
<td>118201</td>
</tr>
<tr>
<td>cagri/hipermarket</td>
<td>Danone MilkShake Çilek-Vanilya 220 ml</td>
<td>4.25</td>
<td>4566</td>
<td>market</td>
<td>danone.adet</td>
<td>milkshake cilek vanilya</td>
<td>danone milkshake çilek vanilya 220.0 millilitre</td>
<td>220</td>
<td>millilitre</td>
<td>danone</td>
<td>116110</td>
</tr>
</tbody>
</table>
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Machine Learning Approach

- scikit-learn library has been used.
- Some preprocessing steps have been experimented. (TF-IDF, lemmatization etc.)
- Scikit-learn codes were executed on local computers. (by fetching data from cluster to a local computer)

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic Regression</td>
<td>0.96</td>
<td>0.94</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>Support Vector Machine</td>
<td>0.94</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.92</td>
<td>0.83</td>
<td>0.79</td>
<td>0.80</td>
</tr>
</tbody>
</table>
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Machine Learning Approach (What’s next)

- sparknlp and spark.ml libraries are being planned to use.
- Ensemble learning models such as XGBoost may be used in next.
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Deep Learning Approach

- 3 HuggingFace models have been tried. (dbmdz/bert-base-turkish-cased, dbmdz/bert-base-turkish-128k-uncased and bert-base-multilingual-cased)

- Codes are implemented via Pytorch and executed in Google Colab environment. (for utilizing GPU)
Project 2: COICOP Assignment to Web Scraped Products with ML-based Approach

Deep Learning Approach

<table>
<thead>
<tr>
<th>MODEL</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbmdz/bert-base-turkish-cased</td>
<td>0.91</td>
<td>0.92</td>
<td>0.91</td>
</tr>
<tr>
<td>bert-base-multilingual-cased</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
</tr>
<tr>
<td>dbmdz/bert-base-turkish-128k-uncased</td>
<td>0.93</td>
<td>0.92</td>
<td>0.93</td>
</tr>
</tbody>
</table>
Project 2: COICOP Assignation to Web Scraped Products with ML-based Approach

Deep Learning Approach (What’s next?)

- Using product images that are collected by pollsters, we are planning to fine tune a pre-trained transfer learning models.

- After we constitute our models, we are planning to apply some ensemble learning techniques by utilizing both our text-based and image-based models.
Project 3: Assigning ISCO Codes to Job Advertisements

Background

• Using survey and administrative data may be both time-consuming and require human force.

• These data may not reflect the market demand at that time.

• Our purpose is to build the ml-based ISCO assignment model with labeled training dataset and use this model on daily web scraped data.
Project 3: Assigning ISCO Codes to Job Advertisements

Data

- 16 different web sites
- Approximately 40K job advertisement daily
- Some columns: title, occupation, city, company name, date_poster, gender etc.
**Project 3: Assigning ISCO Codes to Job Advertisements**

**Data**

**Total Job Advertisement Counts by Web Sites**

- careerjet
- cvbenim
- eleman360
- eleman_online
- eleman_uzm...
- eleman_uzm...
- elemanhayuzu
- elemannet
- ilangoitr
- isbul
- iskur
- kariyernet
- linkedin
- secrecy
- toptalent
- unisbul
- yenbiris

![Bar Chart](chart.png)
Project 3: Assigning ISCO Codes to Job Advertisements

Machine Learning Approach

- The machine learning model was trained by using Apache Spark ml library on big data cluster.

- Some Natural Language Processing preprocessing methods (such as N-gram, TF-IDF etc.) have been tried.

- Support Vector Machines and Logistic Regression models have been used at this stage.
## Project 3: Assigning ISCO Codes to Job Advertisements

### Machine Learning Approach

<table>
<thead>
<tr>
<th>Method</th>
<th>Class count</th>
<th>Job adv. Count for each ISCO code</th>
<th>F1-score</th>
<th># of rows</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>32</td>
<td>&gt;100</td>
<td>%88.4</td>
<td>6501</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>32</td>
<td>&gt;100</td>
<td>%83</td>
<td>6501</td>
</tr>
<tr>
<td>SVM</td>
<td>77</td>
<td>&gt;50</td>
<td>%82.91</td>
<td>9616</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>77</td>
<td>&gt;50</td>
<td>%75</td>
<td>9616</td>
</tr>
<tr>
<td>SVM</td>
<td>127</td>
<td>&gt;25</td>
<td>%76.70</td>
<td>11408</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>127</td>
<td>&gt;25</td>
<td>%65</td>
<td>11408</td>
</tr>
<tr>
<td>SVM</td>
<td>210</td>
<td>&gt;10</td>
<td>%62.31</td>
<td>12662</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>210</td>
<td>&gt;10</td>
<td>%57</td>
<td>12662</td>
</tr>
<tr>
<td>SVM</td>
<td>354</td>
<td>&gt;=1</td>
<td>%56.94</td>
<td>13224</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>354</td>
<td>&gt;=1</td>
<td>%56</td>
<td>13224</td>
</tr>
</tbody>
</table>
Project 3: Assigning ISCO Codes to Job Advertisements

Deep Learning Approach

- Some pre-trained models on HuggingFace have been fine-tuned with our dataset on Google Colab environment.

- We have tried different approaches like Word2Vec, BERT etc.

- The best trained model in terms of accuracy have been selected and downloaded to our local system and it is ready to make predictions.
Project 3: Assigning ISCO Codes to Job Advertisements

Deep Learning Approach
Project 3: Assigning ISCO Codes to Job Advertisements

Deep Learning Approach

<table>
<thead>
<tr>
<th>Class count</th>
<th>Job adv. Count for each ISCO code</th>
<th>F1-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>22</td>
<td>&gt;=1000</td>
<td>%92</td>
</tr>
<tr>
<td>76</td>
<td>&gt;=500</td>
<td>%83</td>
</tr>
<tr>
<td>154</td>
<td>&gt;=250</td>
<td>%76</td>
</tr>
<tr>
<td>222</td>
<td>&gt;=150</td>
<td>%71</td>
</tr>
<tr>
<td>276</td>
<td>&gt;=100</td>
<td>%69</td>
</tr>
<tr>
<td>350</td>
<td>&gt;=50</td>
<td>%64</td>
</tr>
</tbody>
</table>
Thank you for your attendance